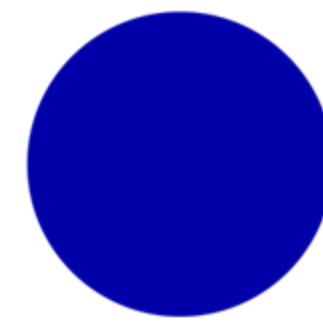


Maquetación Web

Día 1 – Semántica HTML y Maquetación Básica

Índice

1. Diferencias entre Front-end y UX/UI
2. Tipos de proyectos web, W3C y código libre
3. Estructura HTML, semántica y SEO
4. Listas y formularios HTML
5. Introducción al CSS
6. CSS básico
7. CSS Flex-box
8. Glosario
9. Retos



Front End, UX y UI

DEFINICIÓN

UX

UX es literalmente toda experiencia que ocurre cuando uno interactúa con un producto o servicio, sea digital o no.

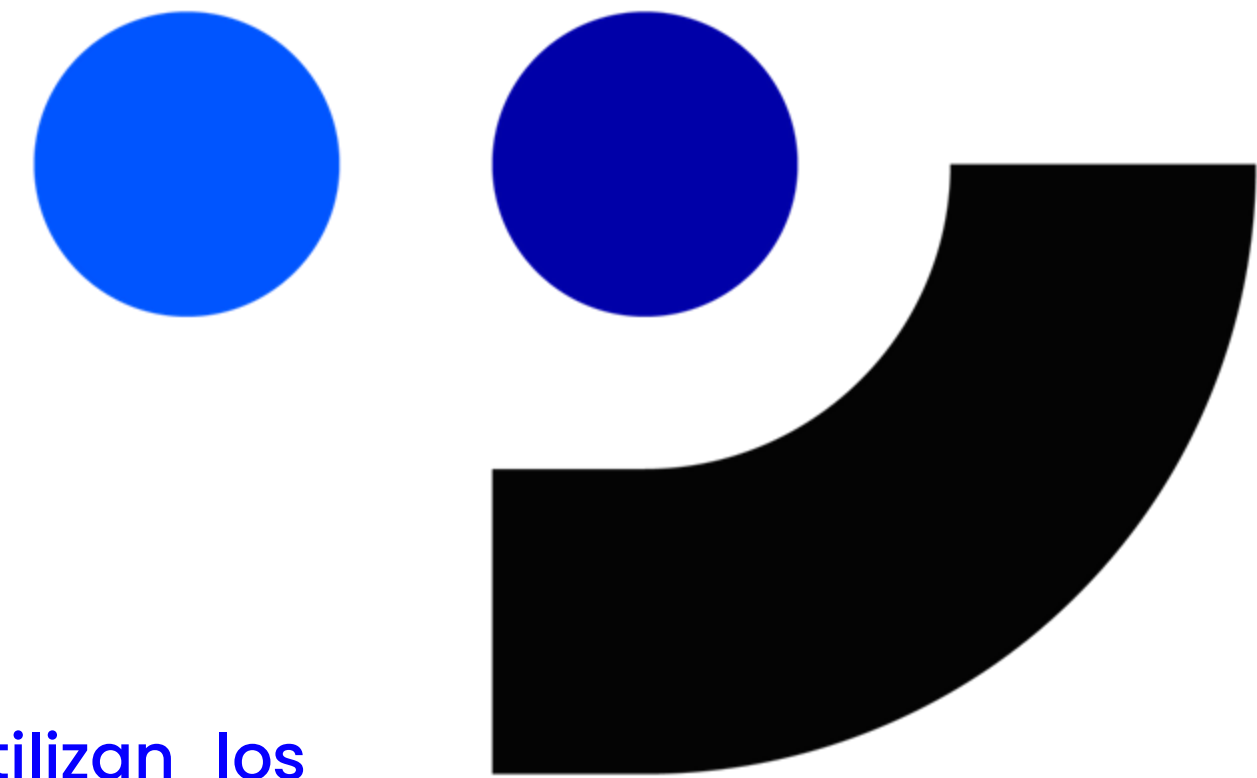
Trabaja sobre el diseño , tratando de amoldarlo a las personas/ clientes y a las necesidades que tienen.

Los diseñadores UX son los encargados de hacer los diseños fáciles, útiles, valiosos, accesibles e intuitivos.

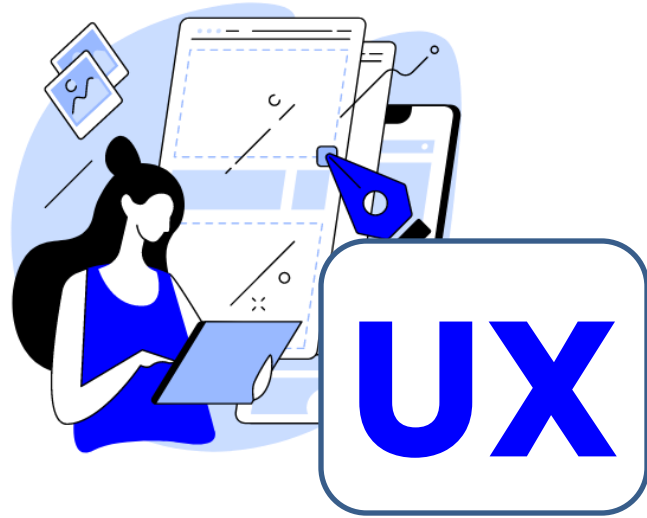
UI

UI es el proceso que utilizan los equipos de diseño para crear productos que brindan, experiencias significativas. Esto implica todo el proceso del diseño del producto, incluidos los aspectos de marca, diseño usabilidad, funcionalidad etc.

Un UI Designer se encarga de crear visualmente la interfaz del usuario para que vaya acorde a la experiencia del usuario.



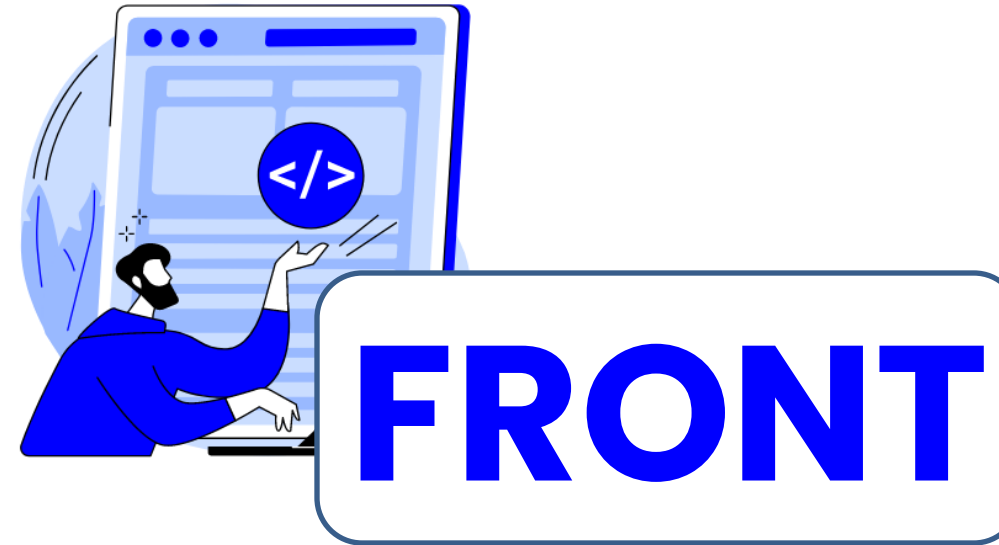
DEFINICIÓN



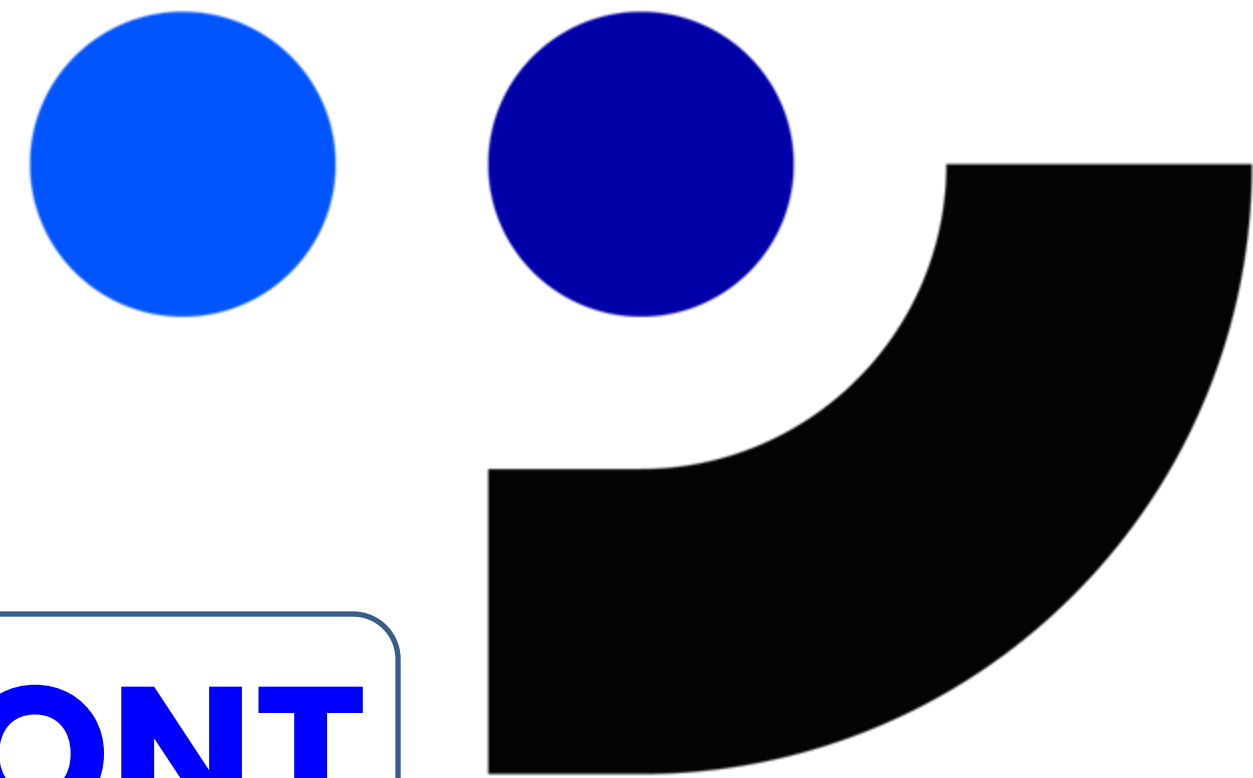
- Investiga las **necesidades de usuario** para mejorar la usabilidad, los flujos de usuario, y las conversiones.
- Se preocupa por la arquitectura de la información y accesibilidad.
- Hacen revisiones del producto y proponen mejorar en función al análisis de datos.



- **Diseña las interfaces, mockups y prototipos**, así como las pantallas y las interacciones de una aplicación.
- Aplica patrones de diseño para crear, layouts, botones, formularios etc.
- Crea guías de estilo que se usan en toda la aplicación, asegurando consistencia y familiaridad para el usuario.



- **Se encarga de implementar correctamente el diseño en código**, maximizar el rendimiento de carga y lograr el correcto funcionamiento de la página, manteniendo la experiencia de usuario y diseño dados por los equipos de UX y UI.
- Trabajan con lenguaje como HTML, JavaScript o Css.
- Utilizan frameworks como Angular o React.

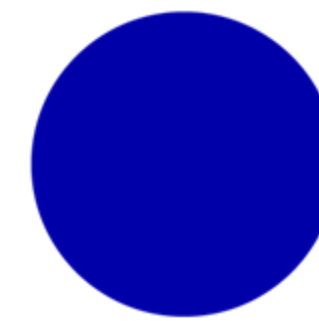


Tipos de proyectos web,

¿QUÉ ES?

SEO

- SEO son las siglas de **Search Engine Optimization** (Optimización de Motores de Búsqueda)
- Conjunto de reglas, estrategias y técnicas de optimización de una página web.
- Gracias al SEO la página web puede **obtener un buen posicionamiento** en los buscadores de Internet como Google, Yahoo o Youtube.
- Básicamente SEO un conjunto de acciones orientadas a mejorar el posicionamiento de un sitio web en la lista de resultados de Google, Bing y otros buscadores de internet
- Guía Completa de SEO: <https://rockcontent.com/es/blog/que-es-seo/>

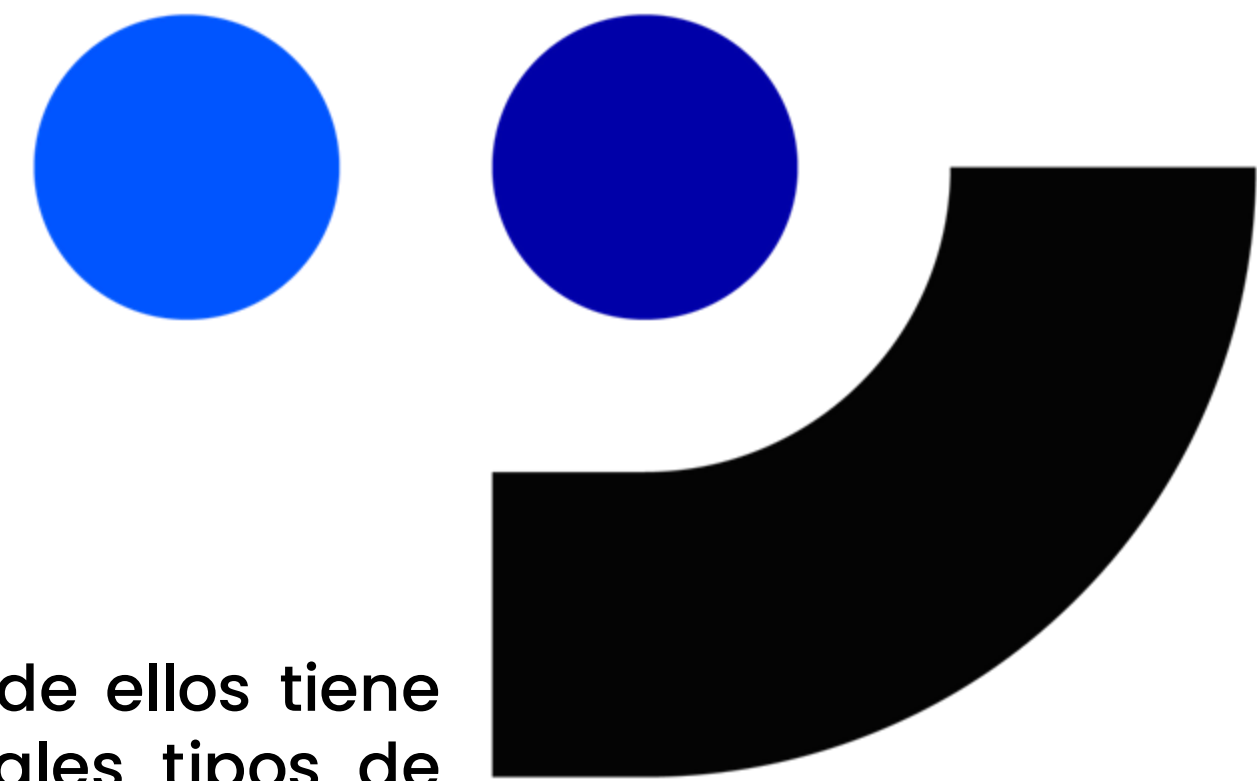


DEFINICIÓN

TIPOS DE PROYECTOS

Existen muchos tipos de proyectos web y evidentemente cada uno de ellos tiene objetivos diferentes. Es por esto que vamos a conocer los principales tipos de proyectos que existen en la actualidad.

- [Landing page](#): Tiene como objetivo informar y vender un producto, deberemos poner atención a la estética y al SEO para su posicionamiento en los buscadores.
- [Aplicación web](#): Es una herramienta que realiza funciones complejas para ofrecer un servicio web. La parte más importante de la maquetaación será el UX y con el objetivo de facilitar al usuario el uso de esas funciones complejas de la formas mas sencilla posible.
- [eCommerce](#): Es una plataforma de compra o tienda. Nuestra maquetaación tendrá que captar la atención del cliente, cuidar el SEO y tener un UX impecable en la parte del pago.



¿QUÉ ES?

Stándar y Código Libre

El Consorcio World Wide Web es una comunidad internacional en la que participan las mayores empresas del sector. Estas empresas trabajan conjuntamente entre ellas para desarrollar estándares web actualizados a las nuevas necesidades.

Es por eso que existen las opciones de desarrollo integradas "Dev Tools" en los navegadores como Firefox, Chrome, o Edge.

Con las herramientas de inspección podemos ver todo el código HTML, CSS y JavaScript del lado de cliente de cualquier aplicación.



HTML y SEO

¿QUÉ ES?

HTML Lenguaje de Etiquetas

HyperText Markup Language (HTML) es el lenguaje de etiquetas utilizado para [estructurar el contenido de las páginas web](#).

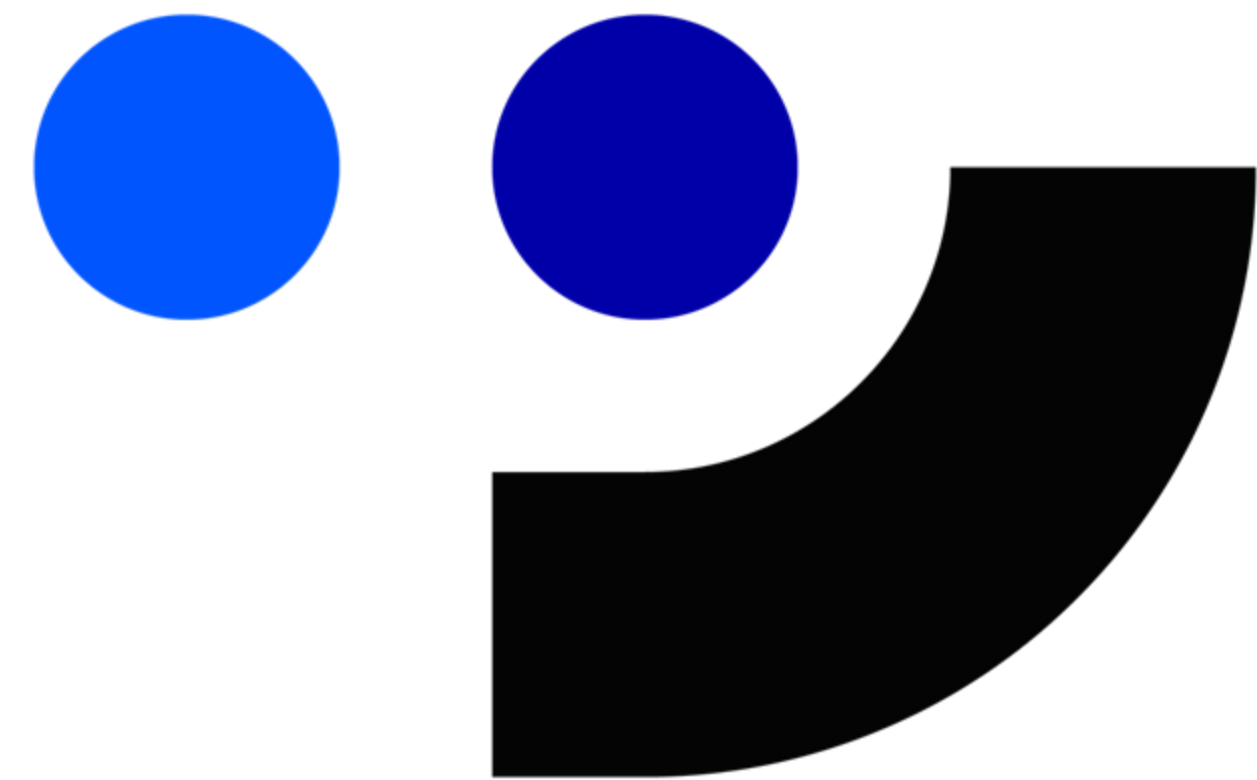
Los lenguajes de etiquetas no poseen capacidad de cómputo ni admite variables a diferencia de los lenguajes de programación.

- **Hypertext** hace referencia a la posibilidad de enlazar distintas páginas web entre sí, se pueden enlazar páginas dentro de nuestro sitio web o enlazar nuestra web a otras páginas externas.
- **Markup** se refiere a las etiquetas, podemos ordenar el contenido y la información que se muestra en la página definiendo las etiquetas adecuadas para su visualización. Además el uso de una buena sintaxis mejorará el SEO y la accesibilidad de nuestra página web.

[HTML: Lenguaje de etiquetas de hipertexto](#)

EJEMPLO

Esqueleto Básico HTML



```

2
3  <!-- con el valor !DOCTYPE definiremos la version HTM que queremos usar para nuestro archivo -->
4  <!-- y con el valor html dedidimoas la última versión disponible del estandar -->
5  <!DOCTYPE html>
6  <!-- Con el atributo lang en la etiqueta HTM definiremos el idioma para todo el documento -->
7  <html lang="en">
8      <!-- La etiqueta head contendra todos los metadatos o "configuración" del documento -->
9  <head>
10     <!-- Definición del formato de codificación de caracteres -->
11     <meta charset="UTF-8">
12     <meta http-equiv="X-UA-Compatible" content="IE=edge">
13     <!-- definimos el valor de las medidas relativas del viewport -->
14     <meta name="viewport" content="width=device-width, initial-scale=1.0">
15     <!-- importación de logo favicon -->
16     <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
17     <!-- importación de hoha de estilos (archivo CSS)-->
18     <link rel="stylesheet" href="main.css">
19     <!-- importación de archivo JS -->
20     <script src="main.js"></script>
21     <!-- titulo -->
22     <title>Document</title>
23 </head>
24 <!-- declaración de la etiqueta body. Marca el inicio de la parte visible por el usuario -->
25 <body></body>
26
27 </html>
28

```

Es el punto de partida de cualquier documento HTML que creemos.

Se puede ampliar la cabecera "head" con todos los metadatos, links y scripts que necesitemos usar en nuestra web.

Es recomendable definir nuestra propia estructura para un mayor control del documento y no depender de interpretaciones.

Los navegadores generan este código automáticamente usando el comando **! + tab**.

EJEMPLO

Body: Semántica para SEO

```

25 <!-- declaración de la etiqueta body. Marca el inicio de la parte visible por el usuario -->
26 <body>
27   <main><!-- Dentro escribimos nuestro código visto por los usuarios -->
28     <header><!-- Define el encabezado de la página -->
29       <div><!-- Divisor genérico, carece de significado semántico -->
30         <h1>Ejemplo semántica</h1><!-- Títulos importantes -->
31       </div>
32     <nav><!-- Define una barra de navegación -->
33       <li><a href="/path">Enlace 1</a></li>
34       <li><a href="/path">Enlace 2</a></li>
35       <li><a href="/path">Enlace 3</a></li>
36     </nav>
37   </header>
38
39   <aside>
40     ...
41   </aside><!-- Define un contenedor lateral -->
42
43   <section><!-- Suele tener varias divisiones en su interior -->
44     ...
45   </section>
46
47   <section>
48     <article><!-- Divisor interior para secciones -->
49     ..
50   </article>
51 </section>
52
53   <footer><!-- Define el pie de página -->
54     <p>Bienvenido <span>Top NochCoder</span></p>
55   </footer>
56 </main>
57 </body>

```

HTML5 incluye nuevas etiquetas estructurales que nos permiten dotar de un mayor sentido a nuestras divisiones semánticas. Estas etiquetas poseen algunas ventajas respecto a el uso de **div**:

- Clarifica el código sin ayuda de comentarios auxiliares, mejorando su mantenimiento.
- Los buscadores entienden mejor el contenido de nuestra página y mejora su posicionamiento (SEO).
- Mejora la **acesibilidad** para los usuarios que utilizan "**screen readers**" o tienen alguna discapacidad visual

DEFINICIÓN

Elementos HTML

Las **etiquetas semánticas** definen la estructura de nuestros documentos, mientras que los elementos poseen propiedades y estilos predefinidos por los navegadores.

Los **elementos** nos permiten dotar de mayor funcionalidad a nuestras páginas y nos permite introducir texto, links, listas, imágenes, vídeos o incluso contenido externo a nuestra web de forma sencilla.

```
<section>
  <h1></h1>
  <h6></h6>
  <p></p>
  <a href=""></a>
  <span></span>
  <li></li>
  <img src="" alt="">
  <svg></svg>
  <iframe src=""></iframe>
  <video src=""></video>
</section>

<!-- Define un título grande -->
<!-- Define un título pequeño -->
<!-- Para insertar un párrafo -->
<!-- Crea un enlace a otra página o sección -->
<!-- Genera una caja con propiedades "inline" -->
<!-- Define un elemento de cualquier tipo -->
<!-- Permite importar imágenes a nuestra página -->
<!-- Genera código svg directamente en la página -->
<!-- Importa contenido de webs externas -->
<!-- Permite importar vídeos a nuestra página -->
```

Los estilos predefinidos dependen del motor de renderizado que tenga cada navegador.

Actualmente los motores más usados son WebKit y Blink.

[Etiquetas atributos y elementos](#)

DEFINICIÓN

Listas en HTML

```
<!-- Listas ordenadas -->
<ol type="A"> <!-- Posibles valores para atributo type: 1, a, A, i, I -->
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
  <li>Elemento 4</li>
  <li>Elemento 5</li>
</ol>
<!-- Listas no ordenadas -->
<ul type="square"> <!--
  Posibles valores para atributo type: disc, circle, square -->
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
  <li>Elemento 4</li>
  <li>Elemento 5</li>
</ul>
<!-- Listas de definiciones -->
<dl>
  <dt>Término 1</dt>
  <dd>Definición 1</dd>
  <dt>Término 2</dt>
  <dd>Definición 2</dd>
</dl>
```

Las listas HTML son **estructuras que nos permiten ordenar el contenido** de nuestra web por puntos o enumerados.

De esta forma podemos agrupar elementos y mejorar el SEO y la accesibilidad.

```
<!-- Combinación de listas -->
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2
    <ol>
      <li>Subelemento 1</li>
      <li>Subelemento 2</li>
      <li>Subelemento 3</li>
    </ol>
  </li>
  <li>Elemento 3
</ul>
```


DEFINICIÓN

Formularios en HTML

```
<form method="" action="">
  <fieldset> <!-- Define subconjuntos dentro de un formulario -->
    <legend>Ejemplo de input</legend> <!-- Etiqueta el elemento fieldset -->
    <label for="inputExample">Soy un input</label> <!-- Etiqueta el elemento input -->
    <input type="text" name="myFirstInput" id="inputExample" list="languages">
      <!-- Define un cuadro de entrada del tipo que se defina en "type" -->
    <datalist id="languages"> <!-- Panel desplegable con sugerencias para el input -->
      <option value="es">Español</option> <!-- Define una opción para datalist -->
      <option value="en">Inglés</option>
    </datalist>
  </fieldset>
  <fieldset>
    <legend>Ejemplo de select</legend>
    <label for="selectExample">Soy un select</label>
    <select name="myFirstSelect" id="selectExample"> <!-- Define un selector de opciones -->
      <option value="es">Español</option> <!-- Define una opción para el selector -->
      <option value="en">Inglés</option>
    </select>
  </fieldset>
  <fieldset>
    <legend>Ejemplo de text area</legend>
    <label for="textAreaExample">Soy un text area</label>
    <textarea name="myFirstTextArea" id="textAreaExample" cols="30" rows="10"></textarea>
  </fieldset>
  <button type="reset">Borrar</button> <!-- Botón para limpiar los valores introducidos -->
  <button type="submit">Enviar</button> <!-- Botón para enviar el formulario -->
</form>
```

Los formularios son el principal punto de **entrada de datos** para luego usarlos en el back-end.

Al ser elementos tan delicados y de interacción con el usuario, debemos darle una especial importancia a su correcta implementación.

Los atributos principales de la etiqueta **form** son:

- **Method:** Especifica el verbo http responsable de la comunicación entre el cliente y el servidor.
- **Action:** Especifica la url dónde se procesará la petición del servidor.

Metodología BEM

DEFINICIÓN

Metodología BEM

BEM CSS es una **metodología de nomenclatura para definir las clases en los nodos HTML** del documento. Es decir, es una manera de nombrar las clases de los nodos de tu HTML para posteriormente atacarlos con **CSS** de una manera fácil, sencilla y clara.

Su objetivo es dar mucha más **transparencia y claridad en tu estructura HTML y CSS**.

Esta metodología nos dice cómo se relacionan las clases entre sí, lo que es particularmente útil en secciones complejas del documento. Y se divide en:

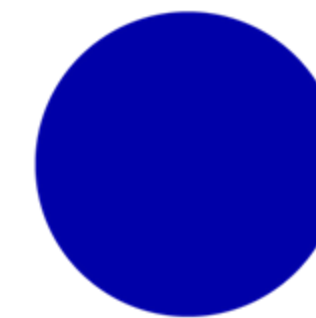
BEM

`bloque__elemento--modificador`

- **Bloques:** son una **sección independiente que tiene significado propio por sí solo**. Contiene todos los nodos HTML de una estructura a la que te estés refiriendo.
- **Elementos:** son **porciones más pequeñas internas a un bloque**. Se usa para ir dividiendo el bloque en segmentos más pequeños.
- **Modificadores:** sirven para **modificar algunas propiedades de un bloque o elemento**.

DEFINICIÓN

Metodología BEM



```

62
63
64
65
66 <body>
67   <div class="item item--modifier">
68     <div class="item__element">Elem 1</div>
69
70     <div class="item__element">
71       <div class="item__another-element">Elem 2</div>
72       <div class="item__another-element--modifier">Elem 3</div>
73     </div>
74
75     <div class="item__element item__element--modifier">Elem 4</div>
76
77   </div>
78
79 </body>
80
81
82
83
84
85
86
87

```

```

2
3 .item{
4   color: inherit;
5   padding: 10px;
6 }
7
8   .item--modifier{
9     background-color: black;
10  }
11
12   .item--modifier2{
13     background-color: brown;
14  }
15
16 .item__element{
17   color: blue;
18   padding: 5px;
19 }
20
21   .item__element--modifier{
22     color: red;
23   }
24
25 .item__another-element{
26   color: violet; }
27
28   .item__another-element--modifier{
29     color: yellow; }
30

```

Elem 1

Elem 2

Elem 3

Elem 4

Introducción a CSS

¿QUÉ ES?

Introducción a CSS

CSS es la abreviatura de "cascading style sheets" y nos [permite definir estilos](#) en nuestros proyectos web.

Como su nombre indica es [una hoja de estilos en cascada](#), lo que quiere decir que si definimos un estilo al principio de nuestro documento y después volvemos a definirlo con valores diferentes, este último estilo [sobrescribirá al anterior](#).

CSS es uno de los pilares en la creación de páginas web, es una potente herramienta que nos permite controlar tanto el diseño como modificar la estructura del documento.

Podemos decir que CSS es como un [potenciador del código HTML](#) que aplica estilos a nuestros documentos de una forma más estructurada y con mejor mantenimiento.

Podéis consultar más información en la [documentación oficial](#).



¿QUÉ ES?

Introducción al CSS

Podemos trabajar con CSS de tres formas sobre nuestro proyecto web:

1. Enlazando desde la cabecera un fichero CSS con la etiqueta `<link>`. Siendo esta práctica la más común y considerada como una buena praxis. Esta etiqueta debe estar siempre ubicada entre las etiquetas de `<head></head>`.
2. Podemos inyectar código CSS dentro de la etiqueta `<style></style>`, trabajando con las propiedades CSS directamente.
3. Por último podemos introducir propiedades CSS sobre cualquier etiqueta HTML con el atributo `style=""`. Esta no es una práctica recomendable.

```

84 <!DOCTYPE html>
85 <html lang="en">
86 <head>
87   <meta charset="UTF-8">
88   <meta http-equiv="X-UA-Compatible" content="IE=edge">
89   <meta name="viewport" content="width=device-width, initial-scale=1.0">
90
91   <!-- link de importacion de archivo CSS -->
92   <link rel="stylesheet" href="style.css">
93
94   <!-- Uso de la etiqueta style -->
95   <style>
96     .header{
97       color: blue;
98       background: black;
99     }
100  </style>
101  <title>mi Documento</title>
102 </head>
103
104
105 <body>
106   <!-- uso del atributo style -->
107   <div style="border: 1px solid blue;">
108     <p style="color: blue;">
109       Bienvenidos al Bootcamp de desarrollo Web y Móvil.
110     </p>
111   </div>
112 </body>
113 </html>
  
```

DEFINICIÓN

Selectores Simples

CSS puede trabajar sobre HTML a través de los selectores, estos nos permiten indicar a qué elemento/s queremos aplicar nuestros estilos.

Existen cuatro tipos de selectores básicos:

- Selectores de **etiquetas**: `div`, `p`, `section`, etc.
- Selectores de **clases**: `.myClass`
- Selectores de **identificadores**: `#myId`
- Selectores de **atributos**: `button[type="submit"]`

```

80
81
82
83
84
85
86 <!-- ejemplo selectores -->
87 <body>
88
89   <div>
90     <section class="myClass">
91       <p id="#myId">
92         Bienvenidos al Bootcamp de desarrollo Web y Móbil.
93       </p>
94     </section>
95
96     <button type="submit">Enviar</button>
97   </div>
98 </body>
99
100
101 </body>
102
103
31
32 div{
33   border: 2px solid blueviolet;
34   border-radius: 5px;
35   margin: 10px;
36   width: 60%;
37 }
38
39
40 .myClass{
41   padding: 10px;
42   background-color: pink;
43 }
44
45
46 #myId{
47   font-weight: bold;
48   font-size: 2rem;
49 }
50
51 button[type="submit"]{
52   background-color: black;
53   color: pink;
54   border: none;
55   padding: 20px;
56
57   width: 40%;
58 }
59

```

Bienvenidos al Bootcamp de desarrollo Web y Móbil.

Enviar

DEFINICIÓN

Unidades en CSS

CSS dispone de múltiples unidades de medida para aplicar tamaños y dimensiones a nuestros elementos. Podemos clasificar estas unidades en dos grupos:



Unidades absolutas:

Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia.

Unidades relativas:

Las unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor.

Esto dota a nuestros diseños de cierta flexibilidad para adaptarlo más fácilmente a todas las pantallas.

DEFINICIÓN

Unidades Relativas CSS

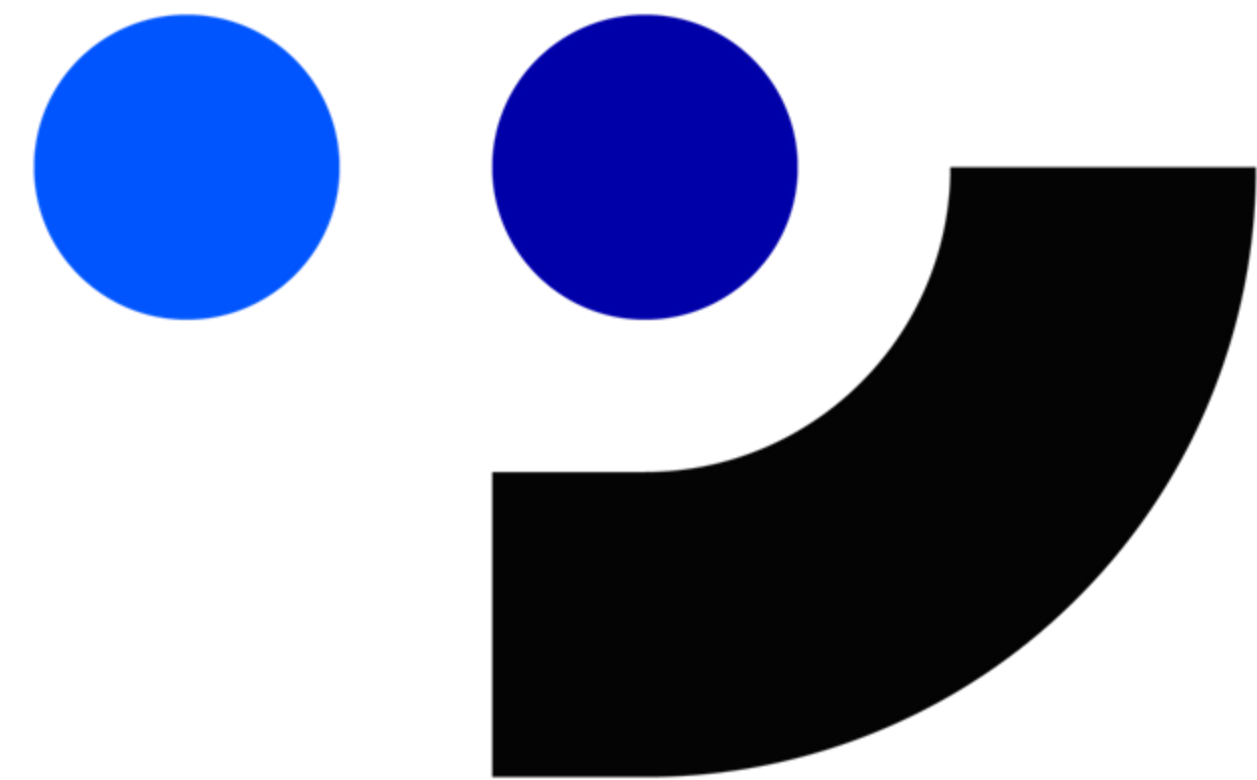
Las unidades relativas se subdividen en 2 grupos:

Relativas a la tipografía

- **Em**: Siempre va a heredar el tamaño de font-size que tenga el contenedor padre.
- **Rem**: Viene dado por el font-size definido en nuestro HTML que por defecto **tendrá** un valor de 16px. Por tanto 1rem = 16px. Si nosotros modificamos el font-size de nuestro HTML, esta equivalencia cambiaría por el nuevo valor dado.
- **Ex**: Aumenta el tamaño de mi letra en cuanto a su altura.

Relativas al viewport (ventana)

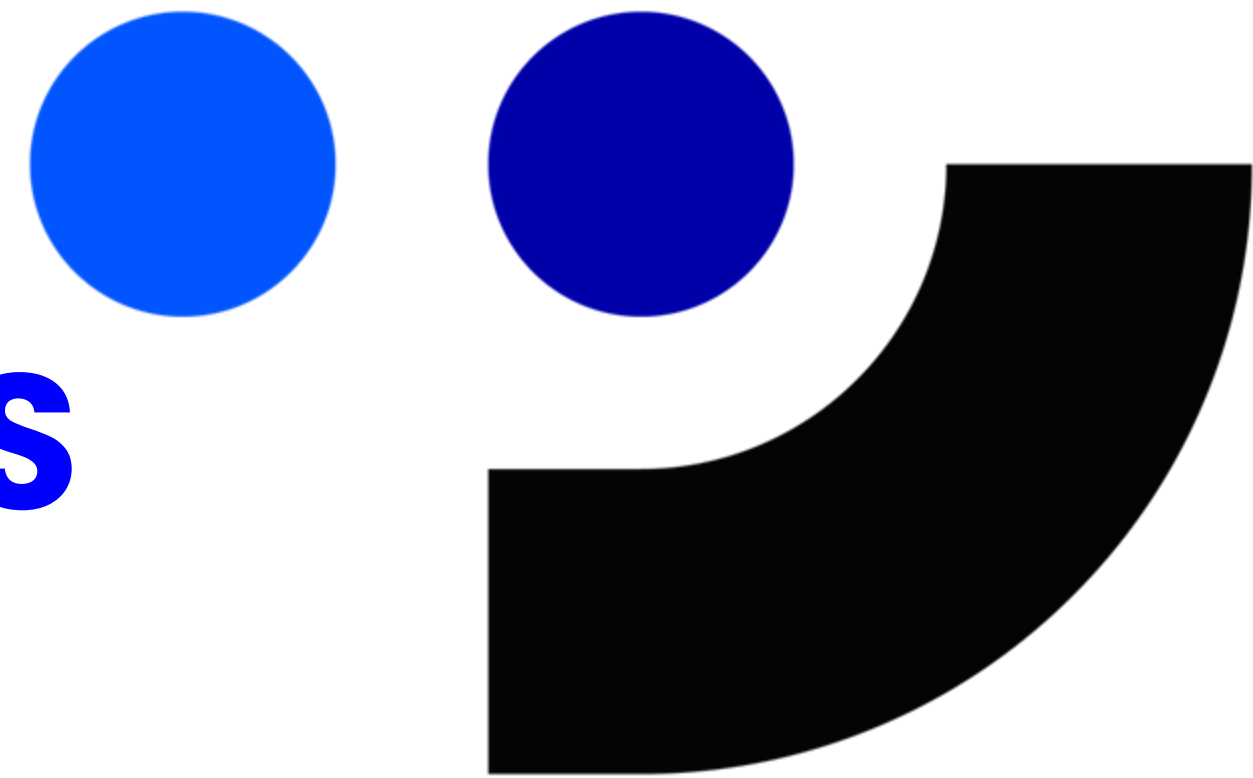
- **Porcentaje (%)**: Relativa respecto a la herencia del elemento padre, si tenemos un padre con una altura de 100px y el hijo tiene una altura de 75%, el hijo tendrá una altura de 75px.
- **vw**: Porcentaje relativo al ancho del viewport.
- **vh**: Porcentaje relativo a la altura del viewport.
- **vmin**: Entre vw y vh toma el que tenga menor valor.
- **vmax**: Entre vw y vh toma el que tenga mayor valor.



DEFINICIÓN

Unidades Absolutas CSS

- **In (pulgadas)**: Una pulgada equivale a 2.54 centímetros.
- **Cm y mm**: Centímetros y Milímetros.
- **Pt (puntos)**: Un punto equivale a $1/72$ pulgadas, es decir, unos 0.35 milímetros.
- **Px (píxel)**: es un punto cuadrado o rectangular que utiliza el lenguaje informático para definir a la parte más pequeña de una pantalla del ordenador o un dispositivo móvil. Por tanto, la imagen que estás viendo es producto del conjunto formado por píxeles, cada uno de un determinado color.



DEFINICIÓN

Propiedades Básicas CSS

CSS posee un enorme abanico de propiedades para crear nuestros estilos que si las utilizamos con ingenio pueden otorgar resultados impresionantes. Estas son algunas de las más básicas.

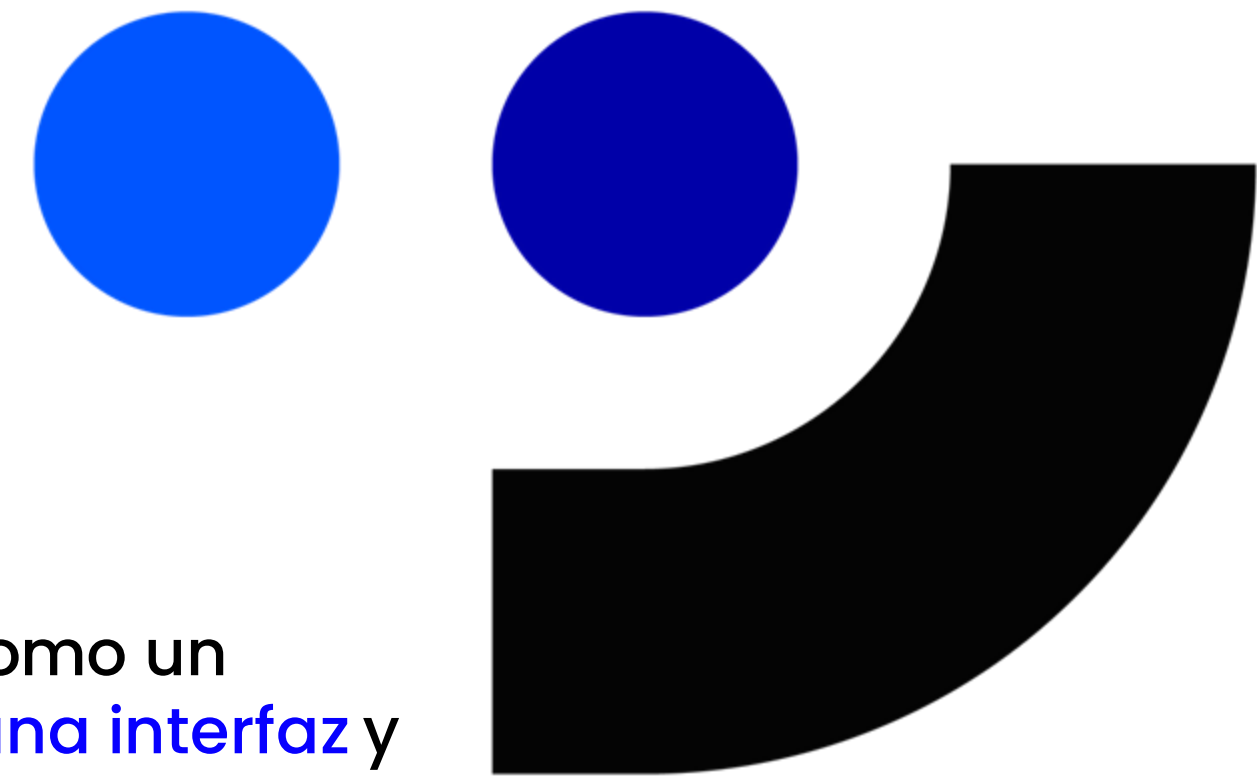
- **Color:** cambia el color de un elemento.
- **Background:** define el fondo de un elemento, puede ser un color, una imagen, un degradado, un mosaico. Background posee varias **sub-propiedades** que facilitan el control de nuestros diseños.
- **Font-size y font-weight:** Definen el tamaño y grosor de la letra. **Son sub-propiedades** la propiedad "font".
- **Margin y padding:** Se utilizan para definir espacios de separación entre elementos. Margin separa las cajas de los elementos entre sí mientras que padding crea espacio entre el borde de la caja y su contenido.
- **Width y height:** Definen el ancho y el alto de las cajas de los elementos.
- **Border:** Define las características del borde de la caja del elemento. Border posee varias sub-propiedades que facilitan el control de nuestros diseños.

CSS Flexbox

DEFINICIÓN

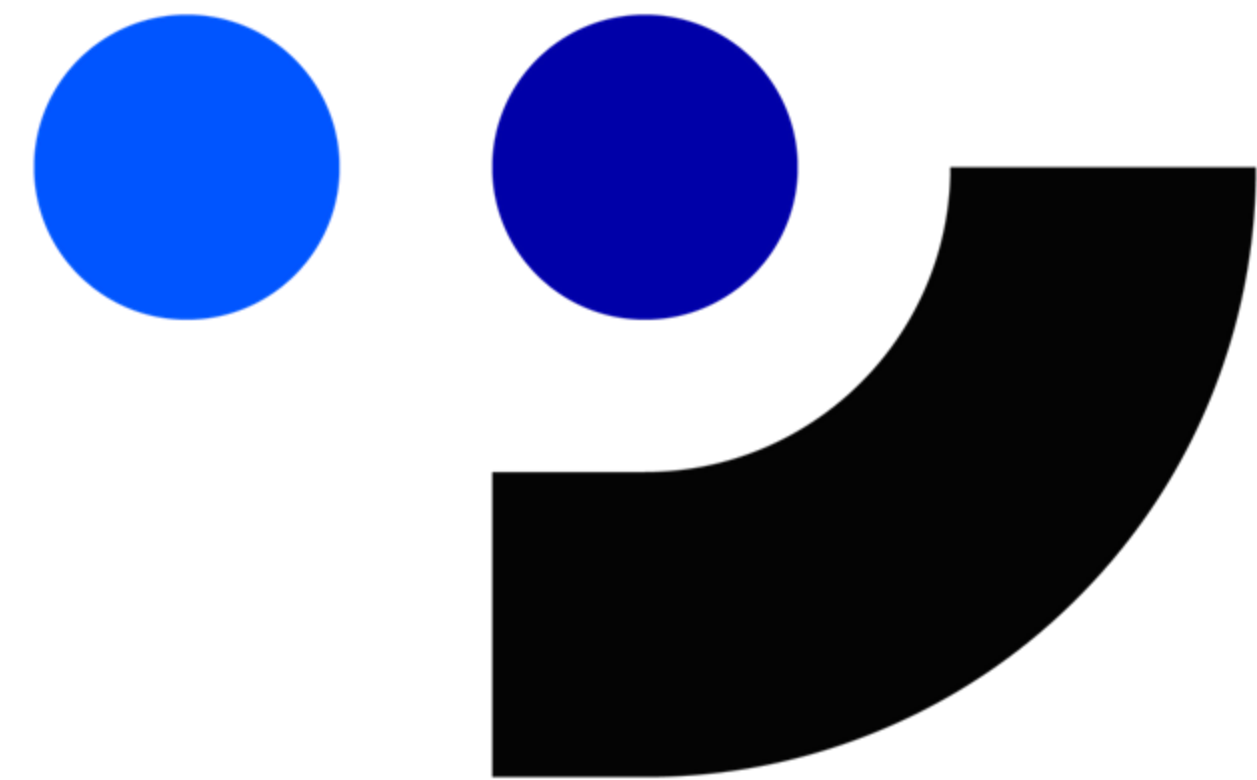
FlexBox

- Flexbox, fue diseñado como un modelo unidimensional de layout y como un **método que puede ayudar a distribuir el espacio entre los ítems de una interfaz** y mejorar las capacidades de alineación.
- **Flexbox** es **unidimensional** es decir, flexbox **maneja** el **layout en una sola dimensión a la vez**, ya sea como fila o como columna.
- Para entender bien el funcionamiento de flex tenemos que tener en cuenta que su modo de trabajo es mediante relaciones **padre-hijo**. Padre-hijo es un tipo de relación que existe entre los elementos del documento, donde un elemento es hijo de otro cuando se encuentra **"encerrado" entre las etiquetas de apertura y cierre del padre**.
- De esta forma un padre puede tener varios hijos pero un hijo solamente puede tener un padre.



DEFINICIÓN

FlexBox



```
103  <body>
104      <!-- contenedor Padre -->
105      <article class="padre">
106          <!-- contenedores Hijo -->
107          <section class="hijo">SECTION 1</section>
108          <section class="hijo">SECTION 2</section>
109          <section class="hijo">SECTION 3</section>
110      </article>
111
112  </body>
```

DEFINICIÓN

Los dos ejes de FlexBox

Cuando trabajamos con flexbox necesitamos pensar en términos de dos ejes, el eje principal y el eje secundario. El **eje principal** está definido por la propiedad **flex-direction**, y el **eje secundario** es el perpendicular a este.

Todo lo que hacemos con flexbox está referido a estos dos ejes, por lo que vale la pena entender cómo trabajan desde el principio.

El **eje principal** que esta definido por flex-direction, que posee cuatro posibles valores:

- row
 - row-reverse
- Si no definimos la propiedad **flex-direction** este tomara el valor row. Si a dicha propiedad le asignamos los valores **row** o **row-reverse**, el eje principal trabajara en base a las filas (es decir en el eje horizontal).
-
- column
 - column-reverse
- Si a la propiedad **flex-direction** en cambio le asignamos los valores **column** o **column-reverse**, el eje principal trabajara en base a las columnas (es decir en el eje vertical).

DEFINICIÓN



Los dos ejes de FlexBox

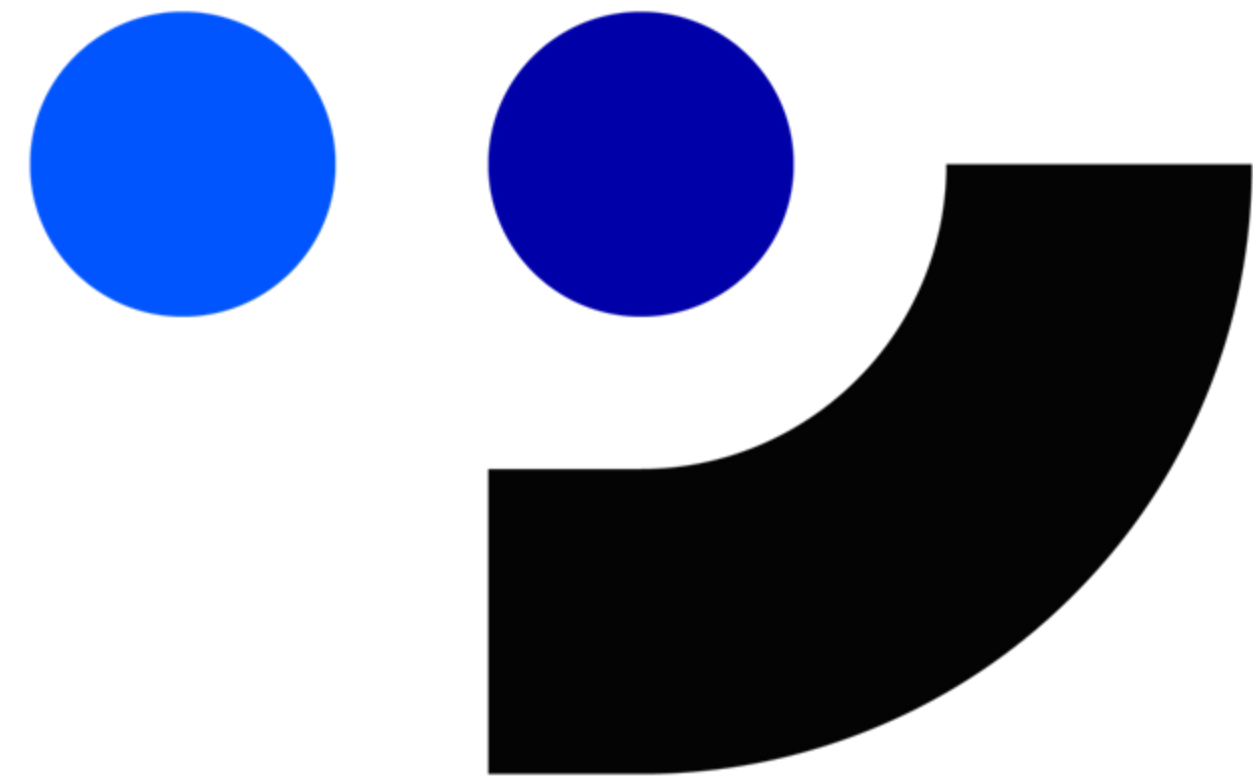
El **eje secundario** siempre será perpendicular al eje principal.

Si el eje principal trabaja con la propiedad flex-direction y los valores row o row-reverse el **eje secundario**, por tanto trabajará en base a las columnas(column).

Si flex-direction (eje principal) es column o column-reverse el eje **secundario** trabajara entonces en base a las filas(row).

Tomemos el caso en que definimos la propiedad **flex-direction:row**

EJE PRINCIPAL:  eje horizontal (fila)
EJE SECUNDARIO:  eje vertical (columna)



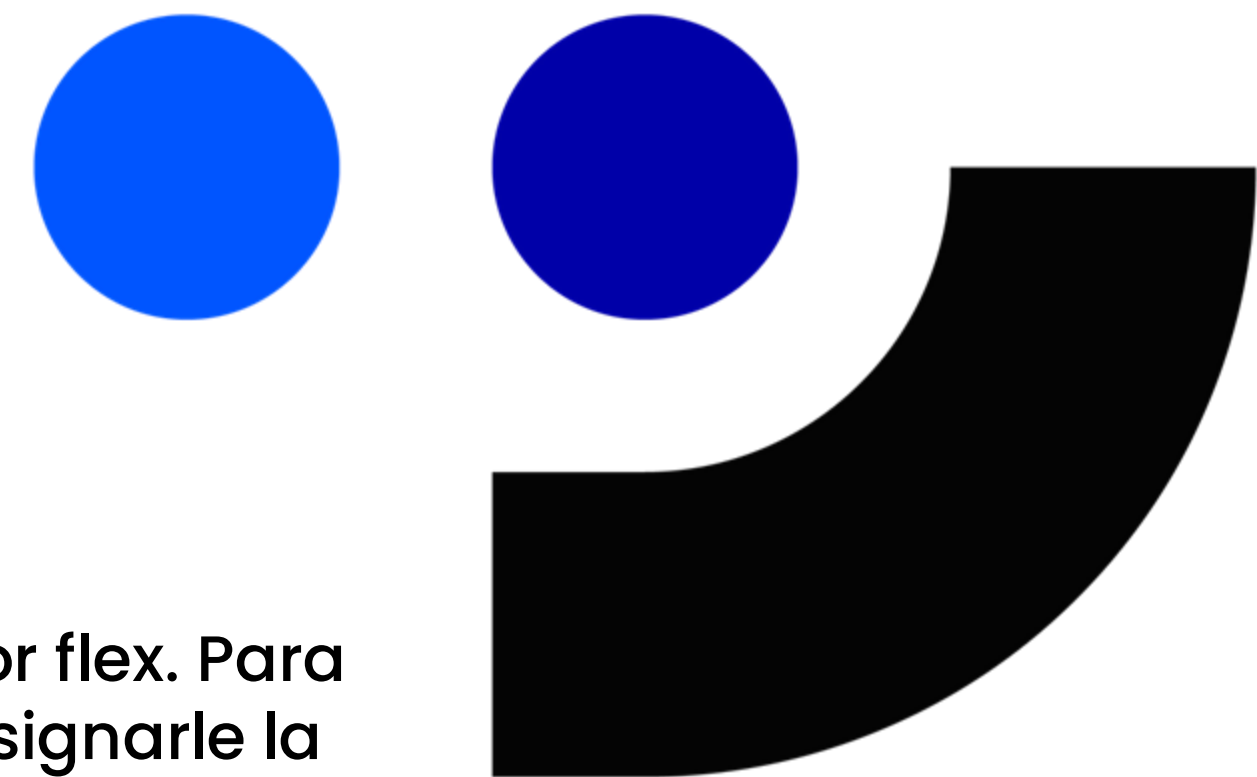
DEFINICIÓN

El contenedor FlexBox

Un área del documento que contiene un flexbox es llamada contenedor flex. Para crear un contenedor flex, simplemente al contenedor padre debemos asignarle la [propiedad display como flex](#).

Tan pronto como hacemos esto, los hijos directos de este contenedor padre se vuelven ítems flex.

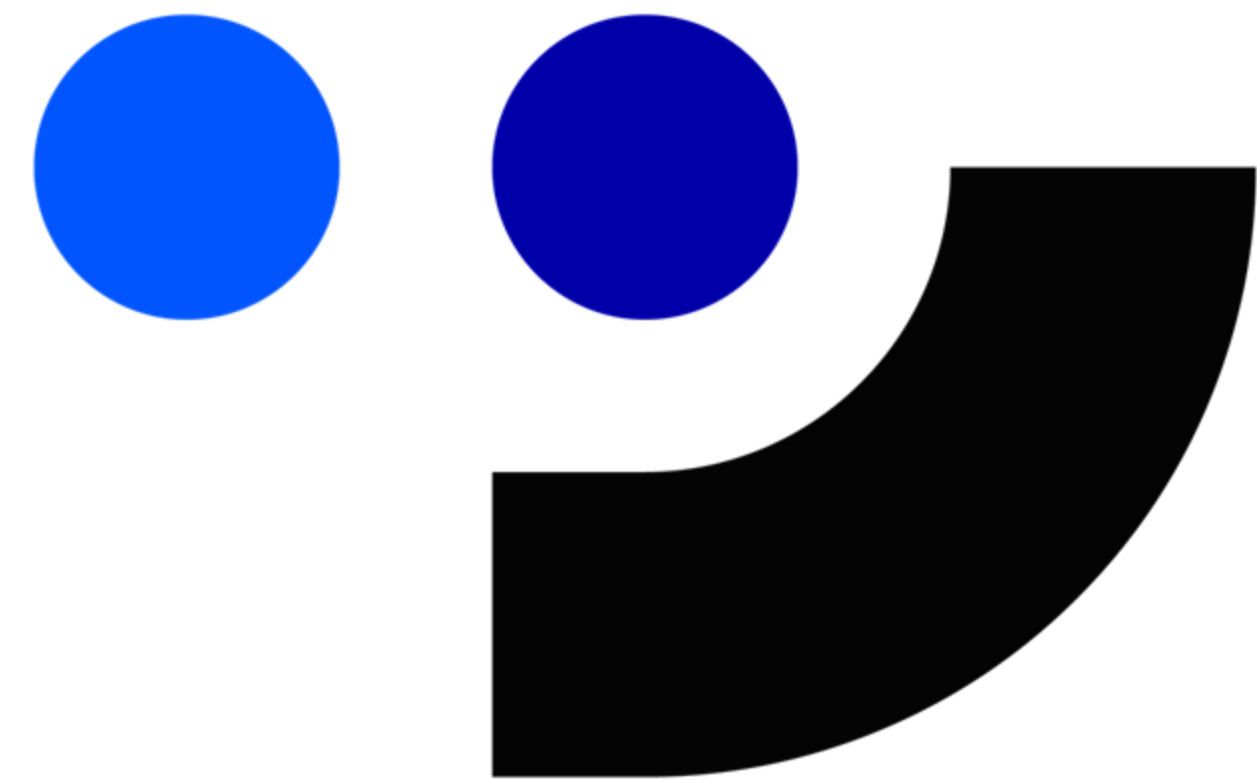
- **flex-direction:** [define el eje principal](#), por defecto viene definido como row (eje horizontal) y se le puede asignar el valor de row o column.
- **justify-content:** Alinea elementos en [base al eje principal](#), sus posibles valores son:
 - **flex-start:** Alinea elementos al inicio del contenedor.
 - **flex-end:** Alinea elementos al final del contenedor.
 - **center:** Alinea elementos en el centro del contenedor en base al eje principal.
 - **space-between:** Posiciona el primer elemento al inicio y el ultimo al final, ubica todos los elementos intermedios con la misma distancia entre ellos.
 - **space-around:** Distribuye los elementos dejando el mismo espacio alrededor de ellos (izq/dcha)
 - **space-evenly:** Distribuye elementos con la misma separación solo entre contenedores hermanos.



DEFINICIÓN

El contenedor FlexBox

- **align-items**: Alinea elementos en **base al eje secundario**.
 - **flex-start**: Alinea elementos a la parte superior del contenedor.
 - **flex-end**: Alinea elementos a la parte inferior del contenedor.
 - **center**: Alinea elementos en el centro en base al eje complementario del contenedor.
 - **baseline**: Muestra elementos en la línea base del contenedor.
 - **stretch**: Elementos se estiran para ajustarse al contenedor.

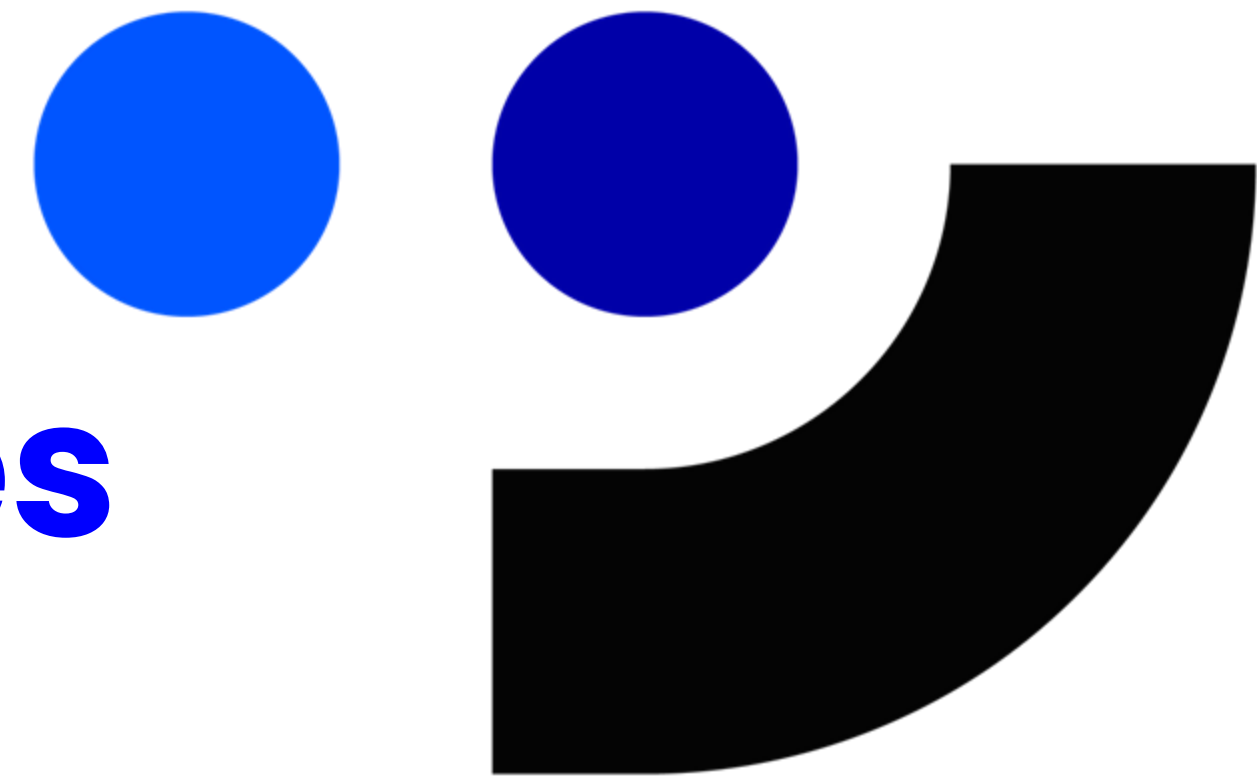


DEFINICIÓN

Propiedades adicionales

Flex también nos brinda otras propiedades muy útiles como:

- **flex-wrap**: Especifica si los elementos "hijos" son obligados a permanecer en una misma línea (reduciendo el tamaño de los elementos hijos) o pueden desplazarse a las siguientes líneas (sin reducir el tamaño de los elementos hijos). Esta propiedad puede tener los siguientes valores : **none**, **wrap**, **wrap-reverse**.
- **align-content**: Esta propiedad nos servirá cuando estemos tratando con un contenedor flex multilinea, que es un contenedor en el que los ítems no caben en el ancho disponible, y por lo tanto, el eje principal se divide en múltiples líneas (usando la propiedad flex-wrap: wrap). De esta forma, **align-content** servirá para **alinear cada una de las líneas del contenedor multilinea**.
- **align-self**: Esta propiedad actúa igual que align-items y sus valores son usados para un elemento hijo en específico y no en el contenedor padre.

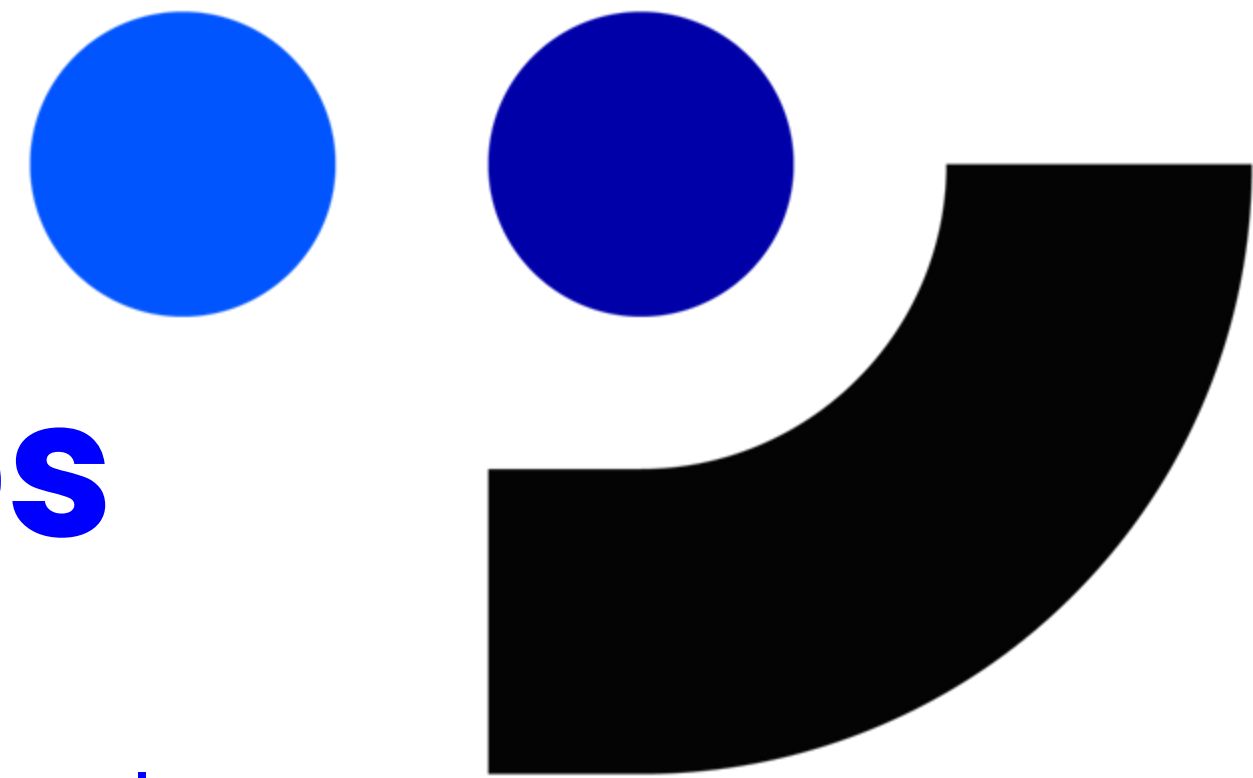


DEFINICIÓN

Propiedades adicionales

Flex también nos brinda otras propiedades muy útiles como:

- **order**: nos permite tener mayor control sobre el orden en el que se ubican los elementos. Por defecto, los elementos tienen un valor 0, pero nosotros podemos usar esta propiedad para establecerlo a un número entero positivo o negativo (entre menor sea el número que asignemos mayor prioridad tendrá y por tanto se ubicará más cerca del inicio).
- **gap**: Es un shorthand que contiene las propiedades **row-gap** y **column-gap**. Estas propiedades, permiten establecer el tamaño de separación entre elementos hijos desde el elemento padre contenedor, y sin necesidad de estar utilizando padding o margin en los elementos hijos.



Glosario

UX/UI

FRONT-END DEVELOPER

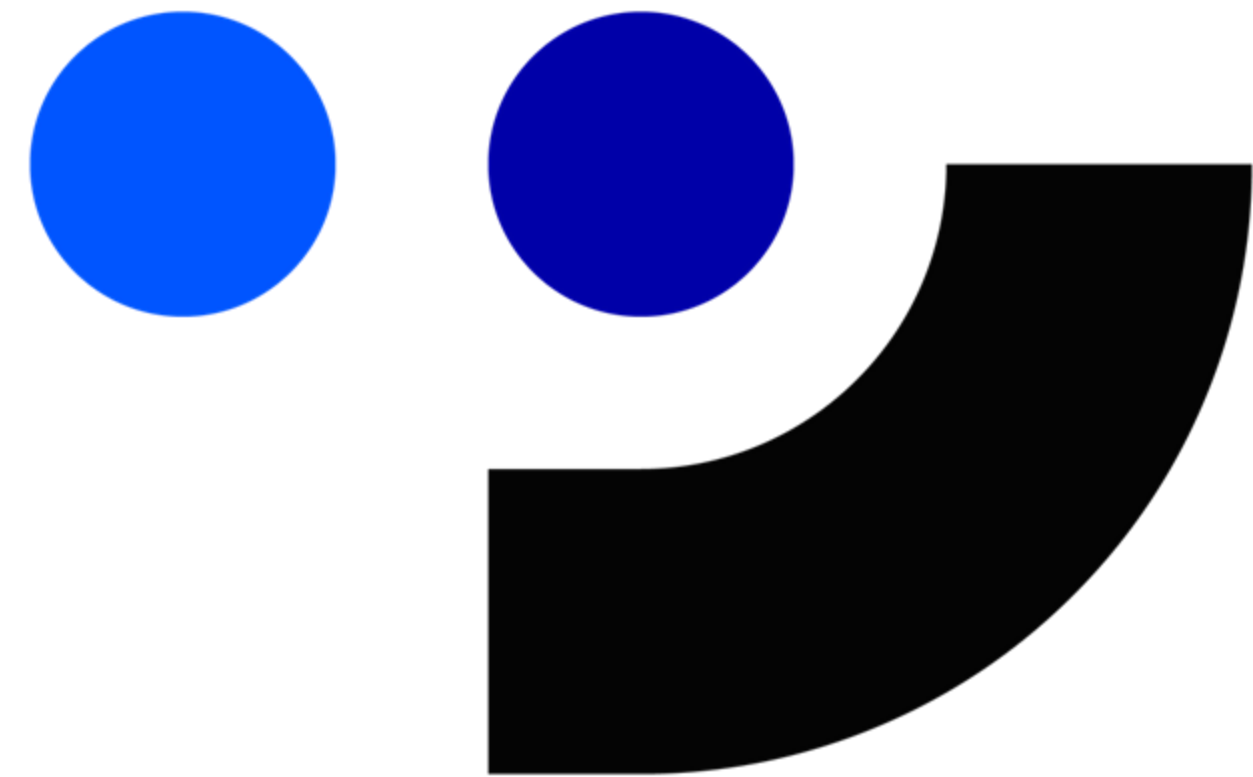
SEMÁNTICA HTML

ELEMENTOS HTML

SELECTORES CSS

PROPIEDADES BÁSICAS CSS

CSS FLEXBOX



Retos



RETOS

Reto 1: Coloca la Rana en la Hoja.

Entra en este enlace <https://flexboxfroggy.com/-es> y completa los 23 niveles.

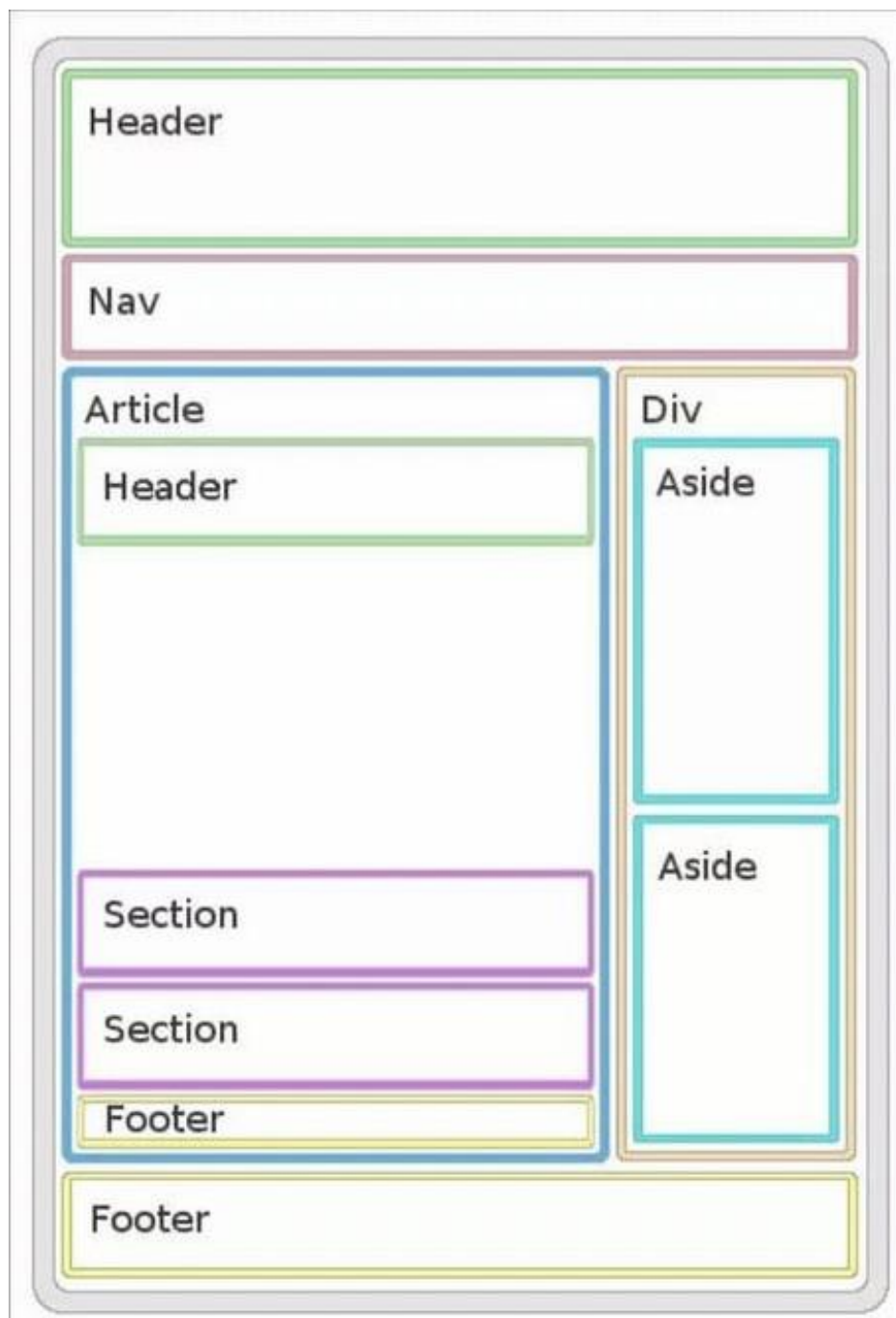
Reto 2: GIT

1. Crea un repositorio de Git llamado "Maquetación". Haz una rama por día(8 días) llamadas "dia1", "dia2" etc..
2. Para los retos de hoy trabajaremos desde la rama "dia1".
3. Crea una carpeta llamada Maquetación_Basica, donde guardaremos todos los retos de hoy.



RETOS

Reto 3: Cajas con Flex-Box



Replica este layout exactamente como se ve en esta imagen.

Los colores y los textos tienen que ser los mismos.

Las proporciones de los contenedores deben ser lo mas parecidas a las que veis en la imagen.

NOTA: Aprovecha lo aprendido de FlexBox para colocar correctamente las cajas. No olvides darle tamaños a los contenedores hijos para que ocupen su lugar, y usar la propiedad: *border Xpx solid colour*, para definir las cajas.

Una vez termines, guárdalo en tu carpeta de Maquetación_Básica y súbelo a git desde tu rama "dia1".

RETOS

Reto 4: Blog Flex-Box

Recrea la página web que verás a continuación, debe verse lo más parecido posible.

La página debe contener tres enlaces en la barra de navegación, que te redirigirán a los distintos apartados.

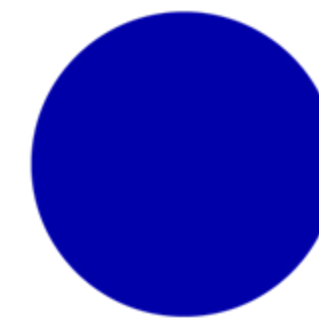
En este reto debes tener muy en cuenta como vas a disponer las cajas en el espacio del que dispones, que cajas van a ser padres y cuales hijos. Usa el truco que has aprendido en el reto anterior de poner borde a esas cajas contenedoras, para que se te más fácil disponer del espacio de forma adecuada.

Utiliza las **propiedades flexbox** para alinear cada uno de los elementos que tengas en tu página. Trata de ser lo mas coherente posible y planifica la disposición basándote el las siguientes diapositivas.

Debes mantener la estructura, de manera que al terminar el reto se vea una cópia exacta.

La barra de navegación debe ser funcional y redirigirte a las distintas páginas.

Cuando termines el reto, guárdalo en la carpeta de "Maquetación_Básica" y súbelo a tu repositorio Git.



RETOS

Reto 4.1. Página Home

Home

Galería

Blog

Bienvenidos a TravelManía

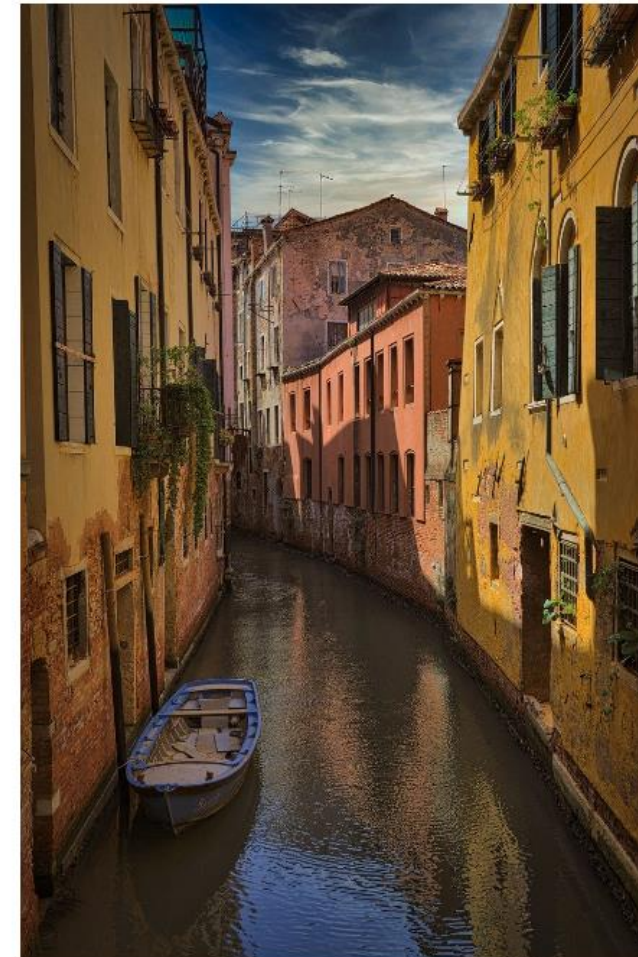
Lorem ipsum dolor sit amet consectetur adipisicing elit. Cum minus libero iure, quos quod cupiditate maxime nemo vero eligendi eaque animi reiciendis earum voluptatem explicabo autem modi, perspiciatis ut pariatur! Lorem ipsum dolor sit amet consectetur adipisicing elit. Vel praesentium earum tempore sit repellendus blanditiis, possimus in placeat fugit ex tenetur molestiae at voluptatem dolores reiciendis id assumenda eaque. Nam!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cum minus libero iure, quos quod cupiditate maxime nemo vero eligendi eaque animi reiciendis earum voluptatem explicabo autem modi, perspiciatis ut pariatur! Lorem ipsum dolor sit amet consectetur adipisicing elit. Vel praesentium earum tempore sit repellendus blanditiis, possimus in placeat fugit ex tenetur molestiae at voluptatem dolores reiciendis id assumenda eaque. Nam!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cum minus libero iure, quos quod cupiditate maxime nemo vero eligendi eaque animi reiciendis earum voluptatem explicabo autem modi, perspiciatis ut pariatur! Lorem ipsum dolor sit amet consectetur adipisicing elit. Vel praesentium earum tempore sit repellendus blanditiis, possimus in placeat fugit ex tenetur molestiae at voluptatem dolores reiciendis id assumenda eaque. Nam!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Cum minus libero iure, quos quod cupiditate maxime nemo vero eligendi eaque animi reiciendis earum voluptatem explicabo autem modi, perspiciatis ut pariatur! Lorem ipsum dolor sit amet consectetur adipisicing elit. Vel praesentium earum tempore sit repellendus blanditiis, possimus in placeat fugit ex tenetur molestiae at voluptatem dolores reiciendis id assumenda eaque. Nam!

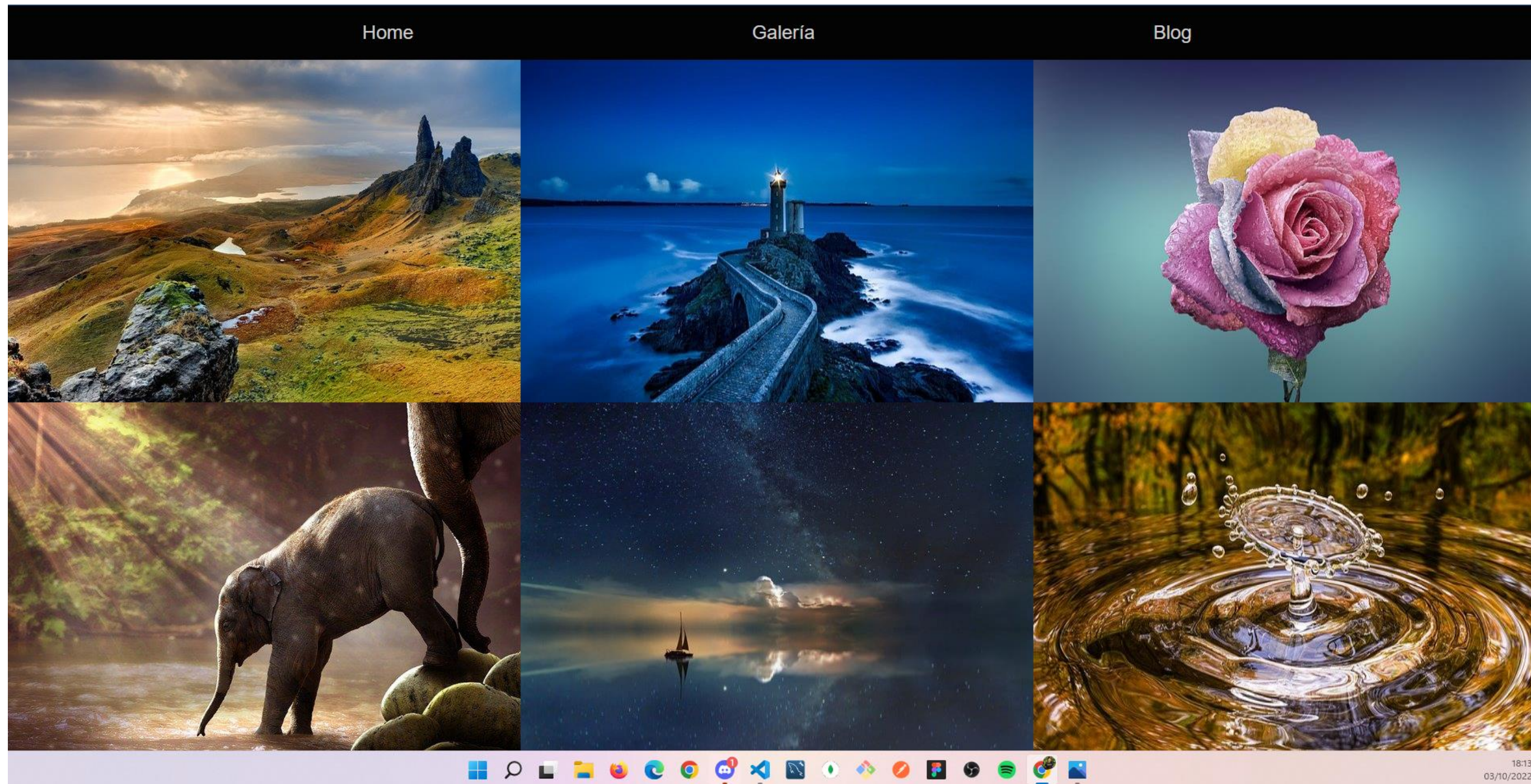
Lorem ipsum dolor sit amet consectetur adipisicing elit. Cum minus libero iure, quos quod cupiditate maxime nemo vero eligendi eaque animi reiciendis earum voluptatem explicabo autem modi, perspiciatis ut pariatur! Lorem ipsum dolor sit amet consectetur adipisicing elit. Vel praesentium earum tempore sit repellendus blanditiis, possimus in placeat fugit ex tenetur molestiae at voluptatem dolores reiciendis id assumenda eaque. Nam!



RETOS

Reto 4.2. Página Fondos

NOTA: Las imágenes en esta página deben tener el mismo tamaño. Puedes descargarlas [aquí](#).



RETOS

Reto 4.3. Página Blog

Home

Galería

Blog

Un viaje Inesperado



Lorem ipsum dolor sit, amet consectetur adipisicing elit. Iusto magni minus quas sequi sed dignissimos illum exercitationem ut adipisci dicta aperiam, nemo eum, possimus facilis laborum unde, vitae ad aut.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Iusto magni minus quas sequi sed dignissimos illum exercitationem ut adipisci dicta aperiam, nemo eum, possimus facilis laborum unde, vitae ad aut.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Iusto magni minus quas sequi sed dignissimos illum exercitationem ut adipisci dicta aperiam, nemo eum, possimus facilis laborum unde, vitae ad aut.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Iusto magni minus quas sequi sed dignissimos illum exercitationem ut adipisci dicta aperiam, nemo eum, possimus facilis laborum unde, vitae ad aut. Lorem ipsum dolor sit, amet consectetur adipisicing elit. Iusto magni minus quas sequi sed dignissimos illum exercitationem ut adipisci dicta aperiam, nemo eum, possimus facilis laborum unde, vitae ad aut.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Iusto magni minus quas sequi sed dignissimos illum exercitationem ut adipisci dicta aperiam, nemo eum, possimus facilis laborum unde, vitae ad aut.

codenotch